HIGH AVAILABILITY PACKET FORWARDING APPARATUS AND METHOD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit under Title 35 United States Code §119 of United States Provisional Application No. 60/279,099 filed on March 27, 2001.

TECHNICAL FIELD

[0002] The present invention relates in general to routers and packet forwarding engines and, in particular, to an apparatus and method that provides high packet forwarding availability.

BACKGROUND OF THE INVENTION

[0003] Existing router architectures and routing protocols lack certain desirable features. For the purposes of this discussion, router architectures and routing protocols include bridging spanning tree protocols (STPs) as well as routing protocols such as Open Shortest Path First (OSPF) and BGP-4 (Border Gateway Protocol version 4).

[0004] OSPF is a link-state routing protocol. It is designed to be run internally of a single Autonomous System (AS). Each OSPF router maintains an identical database describing the AS's topology. From this database, a routing table is calculated by constructing a shortest-path tree. OSPF recalculates routes quickly in response to topological changes, utilizing a minimum of routing protocol traffic. OSPF provides support for equal-cost multipath. An area routing capability is also provided, enabling an additional level of routing protection and a reduction in routing

protocol traffic. In addition, all OSPF routing protocol exchanges are authenticated.

[0005] BGP-4 is an inter-Autonomous System routing protocol. The primary function of a BGP-4 enabled system is to exchange network reachability information with other BGP-4 systems. The network reachability information includes information about a list of ASs that reachability information traverses. The reliability information is sufficient to construct a graph of AS connectivity from which routing loops may be pruned and certain policy decisions at the AS level may be enforced. BGP-4 also provides a new set of mechanisms for supporting classless inter-domain routing. These mechanisms include support for advertising an Internet Protocol (IP) prefix and eliminates the concept of network class within BGP. BGP-4 also introduces mechanisms that allow aggregation of routes, including aggregation of AS paths. To characterize the set of policy decisions that can be enforced using BGP, one must focus on the rule that a BGP-4 speaker advertises to its peers (other BGP-4 speakers with which it communicates) in neighboring ASs only those routes that it uses itself. This rule reflects the "hop-by-hop" routing paradigm generally used throughout the current Internet.

[0006] It should be noted that some policies cannot be enforced by the "hop-by-hop" routing paradigm, and thus require methods such as source routing. For example, BGP-4 does not enable one AS to send traffic to a neighboring AS with the intention that the traffic take a different route from that taken by traffic originating in the neighboring AS. On the other hand, BGP-4 can support any policy conforming to the "hop-by-hop" routing paradigm. Since the current Internet only uses the "hop-by-hop" routing

paradigm, and since BGP-4 can support any policy that conforms to that paradigm, BGP-4 is highly applicable as an inter-AS routing protocol for the current Internet.

[0007] L3 (layer 3 of the open system interconnection model) routing and bridging protocols were not designed to easily allow dual or synchronous standby architectures within routing switches to provide high availability. Typically, high availability for packet forwarding equipment is achieved through physical duplication of switches. Physical duplication has a high cost due to increased footprint, ongoing management, and cabling costs. It is therefore advantageous to be able to provide a highly reliable and available solution to minimize these costs. Furthermore, physical duplication generally fails to address the most common point of failure in modern packet forwarding equipment, namely software crashes due to errors in program code. Due to the increasing complexity and feature support in modern packet forwarding software, it is difficult to provide software loads that are completely error free. Current packet forwarding systems, however, fail to adequately address detection and failover for software faults.

[0008] High availability for packet forwarding requires a number of features, including: 1) the ability to perform hitless software upgrades; 2) the ability to provide hitless control path failover due to either software or hardware faults; 3) the ability to provide hitless line card failover; 4) the ability to provide hitless path failover, and 5) other features, including synchronization of Routing/Bridging states using database synchronization, which is difficult to provide due to the large amount of state information required to maintain synchronization.

Currently, packet forwarding technology does not support hitless software upgrade or failover, much less the other desirable features listed above.

[0009] It therefore remains highly desirable to provide a means of achieving a high level of packet forwarding availability at a reasonable cost and complexity.

SUMMARY OF THE INVENTION

[0010] It is therefore an object of the invention to provide an apparatus and method for high availability packet forwarding that permits hitless software upgrades and software or hardware failover.

[0011] According to an aspect of the invention there is provided an apparatus including one or more service termination cards having a packet forwarding engine for receiving and forwarding packets. First and second control processors each run a plurality of processes. The first and second control processors are in communication with the service termination cards, and each control processor runs asynchronously with respect to the other. First and second forwarding information bases (FIBs) contain forwarding information maintained by the respective control processors. The service termination cards forward packets in accordance with one of the first and second FIBs depending on an integrity of the processes running on the respective first and second control processors. The integrity of a process is defined as "in-service" or "out-of-service".

[0012] According to another aspect of the invention there is provided a method of ensuring high availability in a packet forwarding process, including steps of operating

first and second control processors independently and asynchronously to generate first and second FIBs. The FIBs are respectively provided to a service termination card. The service termination card operates to forward packets using information from one of the FIBs depending on an integrity of selected processes running on the respective first and second control processors.

[0013] In accordance with an embodiment of the invention, the integrity of the first and second control processors is determined by monitoring selected processes executed by the first and second control processes. This ensures that both hardware and software faults are detected as they occur, and failover process are timely initiated.

[0014] In accordance with an aspect of the invention, full bandwidth utilization is ensured during failover by a bandwidth manager, which releases bandwidth reserved by a failed control processor. The released bandwidth can then be utilized by the operating control processor.

[0015] In accordance with another aspect of the invention, line protection is provided for core MPLS traffic by diversely setup label paths through the two control processors. A FIB manager controls MPLS FIBs so that primary labels switched paths (LSPs) and secondary LSPs are generated by different control processors in each of the first and second FIBs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] Further features and advantages of the present invention will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

[0017] FIG. 1 is a schematic diagram of a computer network including an apparatus in accordance with the invention;

[0018] FIG. 2 is a block diagram of a service termination card shown in FIG. 1;

[0019] Fig 3; is a block diagram of a heartbeat tester shown in FIG. 2 and

[0020] FIG. 4 is a block diagram of a control processor shown in FIG. 1.

[0021] It will be noted that throughout the appended drawings, like features are identified by like reference numerals.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0022] The invention provides an apparatus and method for ensuring high availability for packet forwarding in a packet network. The apparatus has dual control processors that operate asynchronously in parallel to compute separate forwarding information bases (FIBs) selectively used by service termination cards (STCs) for packet forwarding. During normal operation, the STCs use master control processor FIBs for packet forwarding. If integrity of the master control processor is lost, the STCs switch to the FIBs of the alternate control processor. Control processor integrity is determined by the STCs, which send heartbeat query messages to selected software processes running on each control processor. This ensures rapid detection of software and hardware faults in a control processor to improve availability.

[0023] FIG. 1 is a schematic diagram of a computer network **100** that includes a router **102** in accordance with the

invention. The router **102** includes a first control processor (CP0) **104** and a second control processor (CP1) **106**. Each control processor **104,106** can function as a master, or a standby control processor. Each control processor **104,106** creates and maintains a respective forwarding information base (FIB0,FIB1) **108,110**. Each control processor **104,106** is communicatively connected to one or more service termination cards (STCs) **112** by communications busses **109** and **111**, respectively. Each of the STCs **112** is connected by links **114** via an NNI (network to network interface) **115** to a network core **116**. The STCs **112** are also connected **118** to a respective I/O interface **120** that each have a respective Ethernet connection **122** available. The control processors **104,106** are both communicatively connected to an operations, administration and management (OAM) workstation **124**.

[0024] FIG. 2 is a block diagram **200** of an STC **112** shown in FIG. 1. The STC **112** includes a lookup memory **204** that stores the FIB0 **108**, which includes an internet protocol forwarding information base (IP FIB0) **206** and a first multi-protocol label switching (MPLS) primary and backup label switched paths (LSPs) (MPLS FIB0) **210**. The lookup memory **204** also stores the FIB1 **110**, which includes an IP FIB1 **208** and an MPLS FIB1 **212**. The STC **112** further includes a packet forwarding engine **214** that is communicatively coupled at **217A** to the FIB0 **108** and coupled at **217B** to the FIB1 **110**. The STC **112** has a heartbeat monitor **220** communicatively coupled at **222** to the CP0 **104** and CP1 **106** shown in FIG. 1. The function of the heartbeat monitor **220** will be described below with reference to FIGs. 3 and 4.

[0025] Each control processor **104,106** runs all relevant routing protocol applications independently and produces

the relevant tables required for packet forwarding. The forwarding information bases (FIB0 **108** and FIB1 **110**) derived from the respective control processors **104,106** are distributed to all STCs **112** regardless of their interface association. It should be noted that the respective FIBs are not necessarily identical, but are expected to contain the same reachability information. The next hop information will most likely not be the same, however. The packet forwarding engine **214** selects a set of FIBs to use at run-time, typically based on control processor association information. In accordance with an embodiment of the invention, A FIB manager **230** receives FIB information form the respective control processors **104,106** via busses **109,111** and writes the FIB information in the respective FIB0 **108** and FIB1 **110**. The FIB manager **230** is preferably programmed to write the MPLS FIBs so that the primary LSPs of FIB0 are created and maintained by control processor **104**, while the backup LSPs of FIB0 are created and maintained by control processor **106**. In FIB1, the primary LSPs are created and maintained by control processor **106**, while the backup LSPs are written by control processor **104**. Consequently, on transit core network traffic, diversely setup LSPs through the control processors **104,106** permit both line and equipment protection to be achieved in a single router **102** in accordance with the invention.

[0026] During a control processor reset, or software upgrade that causes a control processor **104,106** to go out-of-service, the STCs **112** are informed in a timely manner and switch to use the set of FIBs of the remaining active control processor. Multicast IP packet services, multi-protocol label switching (MPLS), bridging, and various media related protocols use a hot-standby control processor model. In accordance with one embodiment of the invention,

full bandwidth utilization is ensured by a bandwidth manager **240**. The bandwidth manager **240** accepts bandwidth allocation requests from the respective control processors **104,106** via busses **109,111**. The bandwidth manager allocates bandwidth to the respective control processors **104,106**, as required and updates the appropriate FIB information using bus **244** to write to lookup memory **204**. However, if one of the control processors is out-of-service, the bandwidth manager is advised of the control processor's condition by the heart beat monitor **224**, which sends an appropriate signal over connection **242**. On being advised that a control processor is out-of-service, the bandwidth manager releases all bandwidth allocated to the out-of-service control processor, so that it is available to the in-service control processor, which can allocate the bandwidth as required. This permits more efficient failover engineering in the core networks.

[0027] The monitoring of a control processor to determine whether it is in-service is performed by monitoring critical software processes that it runs. The monitoring of an integrity of critical processes running on a control processor **104,106** is described with reference to FIGs. 3 and 4, and is performed by a heartbeat monitor **220**. Integrity of a control processor is defined as being "in-service" or "out-of-service". The heartbeat monitor **220** includes a tables of critical processes **304** that contain a list of selected processes **404** that run on the respective control processors **104,106**. The tables are referenced by a heartbeat inquiry generator **306**, which generates heartbeats **306A**, **306B**, . . . **306C** for respectively monitoring the integrity of process **404A**, **404B** and . . . **404C** running on the control processors **104,106**. The heartbeat monitor **220** sequentially generates and transmits the heartbeat

inquiries **306A,306B,306C** to corresponding processes **404A,404B,404C** (FIG. 4). If each process **404A,404B,404C** returns a heartbeat response **308A,308B, . . .308C** within a predetermined period of time, the integrity of each process is declared "in-service". If any of the processes **404A,404B,404C** fails to return a heartbeat response **308A,308B,308C** within the predetermined period of time, the integrity of that process **404A,404B,404C** and the processor **104,106** that runs it is declared to be "out-of-service".

[0028] The invention supports a hitless control processor reset and also supports hitless software upgrades, as long as the packet forwarding engine **214** on STCs **112** does not require reset.

[0029] It should further be understood that the assignment of routed interfaces is done at the physical or logical interface level. Therefore, given an STC **112**, it is not necessary for all interfaces **224** to be mastered by the same control processor. It should also be noted that a mix of routed and stub routed interfaces to the STCs **112** is permissible.

[0030] All packet services run on both control processors **104,106** at a steady state. This includes both unicast and multicast IP protocols, and all MPLS signaling protocols such as RSVP (reservation protocol) and LDP (label distribution protocol). In general, the control processors are unaware of each other. There are exceptions to this rule, however. First, all local interface (host) routes and subnet routes are injected into the IP forwarding table of both control processors. This exception applies to both UNI **119** and NNI **115** interfaces. Second, for services other than IP unicast, MPLS and RSVP (i.e. services that must be kept

synchronized) software on each control processor **104,106** must be aware of its current state, being a master or a standby, and behaves accordingly.

[0031] At steady state, core-to-core traffic is routed between routed interfaces of the same control processor. Customer-to-core traffic may be distributed between routed interfaces of both control processors **104,106**. Core-to-customer traffic is distributed on ingress between routed interfaces of both control processors. During a control processor reset or software upgrade, the STC **112** is notified that one of the control processors has become unavailable. The STC **112** updates all the logical interfaces to associate with the remaining control processor as the master, which effectively instructs the packet forwarding engine **214** to forward all traffic related to the forwarding table of the remaining active control processor. Note that logic of the packet forwarding engine **214** does not change, and theoretically there should be no packet loss as a result of control processor reset or software upgrade, as long as the packet forwarding engine **214** is not reset. When one control processor is unavailable, core-to- core traffic is routed from any routed interface to the routed interface of the remaining control processor. Customer-to-core traffic is routed towards a routed interface of the remaining control processor.

[0032] IP multicast forwarding tables are downloaded from the master control processor to the STCs **112**. As there is only a single copy of the IP multicast forwarding table on the STC, no decision is required to select which table to use, as is the case with unicast traffic. During a control processor reset or software upgrade, the STCs **112** are notified that a control processor has become unavailable.

Initially, the STC **112** will do nothing, but when one of the following two conditions is met, the FIB manager **230** on the STC **112** will erase all the original content of the forwarding information base. The first condition is met if a timeout has expired. The second condition is met if the new master control processor has re-learnt and distributed the same routes that the FIB manager **230** has already installed.

[0033] IP multicast continues to use the original routes when the master control processor becomes unavailable for a period of time, or after the same set of routes are re-learnt by the new master control processor. A network topology change during control processor switch-over potentially causes packet loss.

[0034] When a control processor **104** (for the purpose of the following discussion designated CPx) becomes unavailable due to a software crash, hardware fault, or for any other reason except a software upgrade (which is discussed below in some detail) all STCs **112** and the other control processor (designated CPy in this example) are notified. The packet forwarding engine **214** adjusts the control processor mastership of all affected interfaces to associate with the remaining control processor CPy. The packet forwarding engine 214 adjusts all UNI Layer 1 and Layer 2 physical and logical port records to associate with the remaining control processor. the packet forwarding engine 214 also adjusts a control processor selection algorithm for any Layer 3 static (stub) interface (either NNI or UNI). CPy operates as if nothing has happened. The Cpy still advertises its own routes, along with all the local routes of the CPx.

[0035] Routing peers of CPx stop hearing from CPx for the duration of the reset. Take OSPF (open shortest path first) as an example, it will take 3 times a single hello time (30 seconds) to detect that CPx is unavailable. Normally, CPx will recover in much less than 30 seconds during a warm restart. Hence, any immediate network wide route flap is minimized. Of course, in the event of a persistent hardware or software failure, routing peers will eventually detect the problem and route around all of the routed interfaces of CPx. After CPx recovers, all routing protocol stacks on CPx restart. CPx then re-establishes peering with its original routing peers. Route flap will occur during this stage for a short period of time. CPx then continues to converge and update its forwarding information bases table on STCs **112**.

[0036] After a predefined period of time, STCs **112** are notified that the CPx forwarding tables are now ready to be used. The packet forwarding engine **214**, control processor mastership of all applicable interfaces, and control processor selection algorithm are reset to their original configuration. The delayed notification of CPx availability to STC **112** is intended to minimize routing loops in the carrier core network while CPx is converging.

[0037] Assumptions relevant to the control processor reset discussion above include: 1) The STCs **112** rely on at least one control processor being operational; and 2) Logical interface configuration and operational status are propagated to both control processors **104,106**, regardless of control processor mastership of the related interface.

[0038] A discussion of how software upgrades are conducted is also relevant to the operation and maintenance of the

apparatus in accordance with the invention. One way of upgrading a general purpose CPU software load requires an upgrade of software on the CP **104,106** as well as on STC **112**. Note that other options for performing software upgrades may exist without departing from the scope of the present invention. Further, although network processor software upgrades may be impacted, they are not described.

[0039] The invention provides a method of performing hitless software upgrades. For example, if a control processor **104** (CPx) is to be upgraded, the CPx is taken out-of-service, but CPy **106** and STC **112** behave exactly as they do in the CP reset scenario described above. The CPx is then reloaded with the new software load and starts running after a reboot. From the perspective of the CPx, all interfaces are down, even though they are still forwarding ingress traffic using CPy's forwarding information bases.

[0040] Each STC **112** is respectively reloaded with the new software version, and restarted. The packet forwarding engine **214** is still forwarding traffic based on CPy's forwarding table. While the STC CPUs are restarting, local services such as loopback detection and MAC address learning are unavailable.

[0041] Following reboot after the new software load, CPx enables its interfaces and establishes peering with neighboring routers. The CPx then downloads its forwarding information bases to the STCs **112**. After a predefined period of time, the STCs **112** switch back to using forwarding information bases of the CPx. CPy is subsequently reloaded with the new software version and reboots to start running. CPy then establishes peering with

neighboring routers. CPy downloads its forwarding information bases to STCs **112**. After running protocol convergence and sanity checks, the STCs 112 switch to using the FIB of CPx, and the software upgrade is complete.

[0042] The invention therefore provides an apparatus and method for high availability packet processing that permits hitless software upgrades and hitless software and hardware fault failover.

[0043] While the preferred embodiments of the invention were described in specific terms, it should be noted that alternative network structures can be similarly utilized within the inventive concepts without straying from the intended scope of the invention. Persons skilled in the art will appreciate that there are other alternative implementations and modifications for implementing the present invention, and that the above implementation is only an illustration of one embodiment of the invention. Accordingly, the scope of the invention is intended only to be limited by the claims included herein.